CSE 322 Socket Programming Online – Name Server

There will be 2 servers – LocalServer and RootServer. Each server has a `Hashtable<String, String>`. Keys of these tables are domain names as String. Values are Ip addresses of those domains.

LocalServer's Hashtable:

```
Hashtable<String, String> table = new Hashtable<>();
table.put("buet.ac.bd", "103.94.135.240");
table.put("uiu.ac.bd", "103.109.52.58");
```

RootServer's Hashtable:

```
Hashtable<String, String> table = new Hashtable<>();
table.put("google.com", "172.217.9.238");
table.put("wikipedia.org", "208.80.154.224");
```

LocalServer: When a client connects to LocalServer, it can send a domain name to LocalServer for IP lookup. If the domain exists in LocalServer's table, LocalServer will send the corresponding IP address back to client. If the domain does not exist in LocalServer, LocalServer will send that domain to RootServer and will wait for RootServer's response. LocalServer will reply to client with whatever response it got from RootServer.

RootServer: RootServer will lookup that domain in its table. If domain exists in RootServer, RootServer will send the corresponding IP Address back to LocalServer. If it does not exist, RootServer will reply "IP Address Not Found".

Client: After establishing connection and receiving ID (myID), client will prompt from user input from console (`Scanner sc = new Scanner(System.in); sc.nextLine()`).

Implementation:

RootServer.java (Run it first, only listen's for LocalServer's connection at PORT 50000. Does not connect to others. No need to create any thread here.)

LocalServer.java (Run it after root, first it connects to RootServer, Then listens for multiple client at PORT 6666. **Treats each client on different thread**).

Client.java (Run it last. No need to create any thread at client side)

Warnings:

- Match strings using ".equals()" method
- Open OutputStream first, always
- Open buffers only one time for each socket